

Explicable Robot Planning as Minimizing Distance from Expected Behavior

Anagha Kulkarni¹, Tathagata Chakraborti¹, Yantian Zha¹,
Satya Gautam Vadlamudi², Yu Zhang¹, and Subbarao Kambhampati¹

Abstract—In order for robots to be integrated effectively into human work-flows, it is not enough to address the question of autonomy but also how their actions or plans are being perceived by their human counterparts. When robots generate task plans without such considerations, they may often demonstrate what we refer to as *inexplicable behavior* from the point of view of humans who may be observing it. This problem arises due to the human observer’s partial or inaccurate understanding of the robot’s deliberative process and/or the model (i.e. capabilities of the robot) that informs it. This may have serious implications on the human-robot work-space, from increased cognitive load and reduced trust in the robot from the human, to more serious concerns of safety in human-robot interactions. In this paper, we propose to address this issue by learning a distance function that can accurately model the notion of explicability, and develop an anytime search algorithm that can use this measure in its search process to come up with progressively explicable plans. As the first step, robot plans are evaluated by human subjects based on how explicable they perceive the plan to be, and a scoring function called explicability distance based on the different plan distance measures is learned. We then use this explicability distance as a heuristic to guide our search in order to generate explicable robot plans, by minimizing the plan distances between the robot’s plan and the human’s expected plans. We conduct our experiments in a toy autonomous car domain, and provide empirical evaluations that demonstrate the usefulness of the approach in making the planning process of an autonomous agent conform to human expectations.

I. INTRODUCTION

Recent advancement in the field of robotics has given us autonomous robots, vehicles, drones, etc. Typically these autonomous systems have the capability to make their own plans which help them achieve their goals. These advances have, naturally, encouraged the possibility of human-robot teaming where the autonomous robots and humans can work alongside each other. However, if the plans that are being generated by the autonomous robots are difficult to comprehend for the human observer, the unexpected behavior from the robot can raise several concerns: it may increase cognitive load, hamper the productivity of the team, and result in safety concerns and distrust towards the robot [1].

This mismatch between the robot’s plans and the human expectations may be explained in terms of difference in the actual robot model and the *human’s understanding of the robot model*. Thus, even with the knowledge of the robot’s

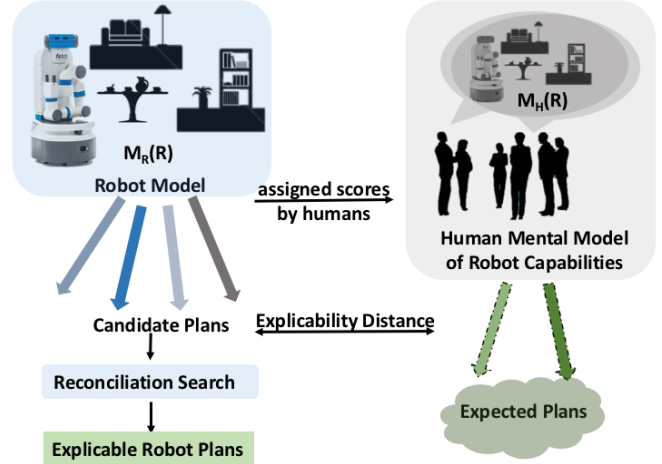


Fig. 1: A schematic diagram of the proposed system. Explicability distance is the plan distance measure between the robot plan and human expected plan. This distance is used to guide the search to generate explicable plans.

goals, it may still not be possible for the human to make sense of the robot’s plan. For example, consider a scenario with an autonomous car switching lanes on a highway. The autonomous car, in order to switch the lane, may make sharp and calculated moves, as opposed to gradually moving towards the other lane. These moves may well be optimal for the car, and backed by the car’s superior sensing and steering capabilities. Nevertheless, a human passenger sitting inside may perceive this as dangerous and reckless behavior, in as much as they might be ascribing the car the sort of driving abilities they themselves have.

To address this issue of the differences between the robot model, $\mathcal{M}_R(\mathcal{R})$, and the human mental model of the robot capabilities, $\mathcal{M}_H(\mathcal{R})$, we develop an approach based on the concept of explicability. An explicable plan is a plan that is generated with the human’s expectation of the robot model; the ability to synthesize explicable plans on the part of the robot thus involves the ability to take into consideration both models into its plan generation process. The intuition is that, the similarity between the robot plan that is generated from the robot model, and the plan that is generated from the human understanding of the robot model determines the explicability of the robot plan. More specifically, the smaller the *explicability distance* between these two plans, the more explicable the robot plan is. Of course, such a

¹Anagha Kulkarni, Tathagata Chakraborti, Yantian Zha, Yu Zhang, and Subbarao Kambhampati are with the Computer Science and Engineering Department at Arizona State University { akulka16, tchakra2, yantian.zha, yzhan442, rao } @ asu.edu

²Satya Gautam Vadlamudi is with Capillary Technologies. vsatyagautam@gmail.com

similarity metric is not readily available, which brings us to the question - *How can a robot learn a distance function between plans that model the notion of explicability, and how can it use this learned similarity model to inform its own deliberative process?* Keeping this in mind, we address the following questions in our paper: 1) Given a domain, can we find an approximation to $\mathcal{M}_{\mathcal{H}}(\mathcal{R})$, the human mental model of the robot capabilities? 2) Can the measures of distance between a robot plan, $\pi_{\mathcal{R}}$, and the plan expected by the human, $\pi_{\mathcal{H}}$, effectively capture the explicability of the robot plan? 3) Can we then integrate the explicability estimates into the robot plan generation process? The outline of our proposed approach is illustrated in Figure 1.

To address the first question, we start out with a robot model, $\mathcal{M}_{\mathcal{R}}(\mathcal{R})$ and generate different robot plans for various initial and goal states. Next, we recruit human subjects and ask them to evaluate these plans by assigning scores to them based on how well they understand them. As a part of the study, the subjects are then asked to answer a questionnaire based on the robot model, in order to elicit implicit human preferences. This questionnaire allows the domain modeler to generate $\mathcal{M}_{\mathcal{H}}(\mathcal{R})$, based on humans assumptions regarding the robot model in that domain. In this paper, we represent both models in PDDL [2], but they can differ in terms of their action representations, preconditions, effects, and costs.

To answer the second question, we explore the relationship between three existing plan distance measures: action set, causal link set and state sequence distances [3], [4] and the plan explicability distance. We use the robot plans, assigned with scores, to determine if the explicability of the plans can be modeled in terms of the aforementioned plan distance measures, in terms of a regression function. For this, we generate the plans expected by the human for the same initial and goal states using the human understanding of the robot model. We then compute the plan distances between the robot plans and human expected plans. We call the function that maps the plan distances to the explicability scores as the *explicability distance*.

To address the third question, we integrate the explicability distance in the search process of the Fast-Downward planner [5]. We perform a cost-bounded anytime search, that can progressively generate more and more explicable plans, using the learned explicability distance as a heuristic guidance. We call this *reconciliation search*. Note that explicability distance exhibits non-monotonicity, i.e. a new action that gets added to a plan prefix can either increase or decrease the explicability distance depending on the context of the plan. We present an analysis on how this property affects our search. For evaluation of our system, we demonstrate the effectiveness of our system in a simulated autonomous car domain, and use human test subjects to evaluate the explicability of the generated robot plans.

II. RELATED WORK

The notion of robots working alongside humans for task achievement has been a popular research direction. It is

challenging, mainly due to the fact that, the robot must consider the human in the loop while making its own decisions. One important requirement for achieving this, is the ability to infer about the human's intent and plan. Various plan recognition algorithms [6], [7] can be applied to perform plan recognition based on a given set of observations as a result of the agent interacting with the environment. After the intent and the plan of the human is identified, researchers have also discussed how the robot can utilize this information while avoiding conflicts [8], [9] or providing proactive help to the human in the loop [10], [11]. There is also work on performing simultaneous plan recognition and generation [12]. However, most of the prior work has only focused on how robots can make plans based on the inferred human intent.

The motivation for generating explicable task plans was first provided in our recent paper [13]. While that work proposes learning explicability as a labeling scheme, in this work, we consider viewing explicability more directly in terms of distances between the plans generated by the robot's own model, and the human's approximation of the robot's model. While explicability focuses on task plans, a related notion of "legibility" has been studied in the context of motion planning [14] and has been shown to be useful in generating socially acceptable behaviors for robots [15], [16].

In most human-robot cohabitation work where robots are proactive agents, it is often assumed that the human model is provided and complete for inferring about the human intent and plan. This is often not true. Although we also assume a human model a priori, our formulation allows us to adjust this model so as to improve model incompleteness (e.g., action preference). There also exists learnable models that do not assume completeness in the first place [17]. Another note is that in [13], [14] and this work, since the model is one level deeper, which is about the robot model from the humans perspective, learning methods are adopted.

III. BACKGROUND

A. Planning

A classical planning problem can be defined as a tuple $\mathcal{P} = \langle M, \mathcal{I}, \mathcal{G} \rangle$, where $M = \langle F, A \rangle$ is the domain model (that consists of a finite set F of fluents that define the state of the world and a set of operators or actions A), and $\mathcal{I} \subseteq F$ and $\mathcal{G} \subseteq F$ are the initial and goal states of the problem respectively. Each action $a \in A$ is a tuple of the form $\langle pre(a), eff(a), c(a) \rangle$ where $c(a)$ denotes the cost of an action, $pre(a) \subseteq F$ is the set of preconditions for the action a and $eff(a) \subseteq F$ is the set of the effects. The solution to the planning problem is a *plan* or a sequence of actions $\pi = \langle a_1, a_2, \dots, a_n \rangle$ such that starting from the initial state, sequentially executing the actions lands the robot in the goal state, i.e. $\Gamma_M(\mathcal{I}, \pi) \models \mathcal{G}$ where $\Gamma_M(\cdot)$ is the transition function defined for the domain. The cost of the plan, denoted as $c(\pi) = \sum_{a_i \in \pi} c(a_i)$, is given by the summation of the cost of all the actions in the plan π . Henceforth, we denote the robot plan as π^R and the human expected plan as π^H .

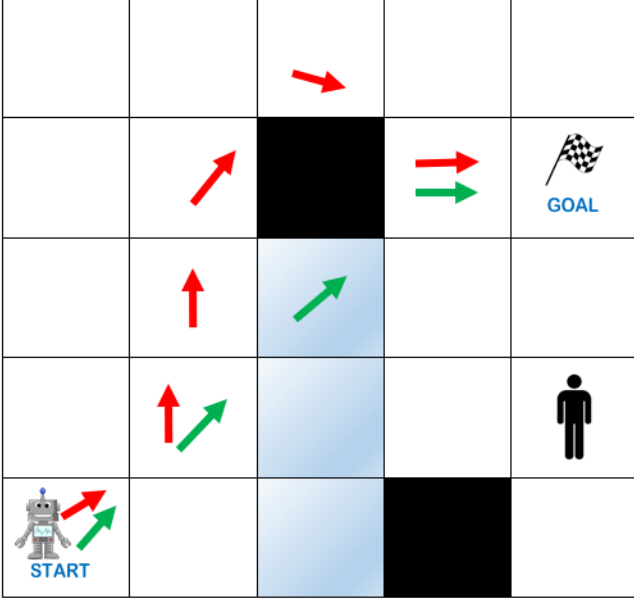


Fig. 2: A simple illustration of how a robot's optimal plan can deviate from the human expectation due to model difference. In this maze, the robot can move in all four direction: up, down, left, right and also diagonally across the grid cells. Some of the cell floors have glass floors and some others have obstacles. The glass floors are harder for the robot to navigate across, because of the reflective surface, it needs to use special sensors which results in an expensive action for the robot. The path in green is the explicable plan whereas the path in red is the robot plan.

B. Plan Distance Measures

We now look at the three plan distance measures introduced in [3] and later refined in [4]. These plan distances are action, causal link and state sequence distances. Although these distance metrics do not satisfy certain mathematical properties [18], they provide a good domain independent measure of the difference between any two plans. Since the goal is to predict the differences in terms of explicability distance between the robot plans and human expected plans, the intuition is that they can be approximated using a combination of plan distance measures that capture different aspects of plans.

1) *Action Distance*: We denote the set of unique actions in a plan π as $A(\pi) = \{a \mid a \in \pi\}$. Given the action sets $A(\pi^R)$ and $A(\pi^H)$ of two plans π^R and π^H respectively, the action distance, δ_a , is computed as the ratio of the actions that are exclusive to each plan to all the actions in the plans [4]. It is written as:

$$\delta_A(\pi^R, \pi^H) = 1 - \frac{|A(\pi^R) \cap A(\pi^H)|}{|A(\pi^R) \cup A(\pi^H)|} \quad (1)$$

This simply means that two plans are similar (and hence their distance measure is smaller) if they contain similar actions. Note that this measure does not take the ordering of actions into account.

2) *Causal Link Distance*: A causal link represents a tuple of the form $\langle a_i, p_i, a_{i+1} \rangle$, where p_i is a predicate variable that is produced as an effect of action a_i and used as a precondition for the next action a_{i+1} . The causal link distance measure is represented similarly to the action distance, by considering the causal link sets $Cl(\pi^R)$ and $Cl(\pi^H)$ instead of action sets described above. It is written as:

$$\delta_C(\pi^R, \pi^H) = 1 - \frac{|Cl(\pi^R) \cap Cl(\pi^H)|}{|Cl(\pi^R) \cup Cl(\pi^H)|} \quad (2)$$

Again, plans are similar, with lower similarity scores, if they have a large number of overlapping causal links.

3) *State Sequence Distance*: This distance measure, as the name suggests, takes the sequences of the states into consideration. This distance captures the context of an action in a given plan. The length of the sequences may differ and therefore there are multiple ways to define this distance measure [4]. We use the representation shown in Eq. 3. Given two state sequences (s_0^R, \dots, s_n^R) and $(s_0^H, \dots, s_{n'}^H)$ for π_R and π_H respectively, where $n \geq n'$ are the lengths of the plans, the state sequence distance is written as:

$$\delta_S(\pi^R, \pi^H) = \frac{1}{n} \left[\sum_{k=1}^{n'} \Delta(s_k^R, s_k^H) + n - n' \right] \quad (3)$$

where $\Delta(s_k^R, s_k^H) = 1 - \frac{|s_k^R \cap s_k^H|}{|s_k^R \cup s_k^H|}$ represents the distance between two states (where s_k^R is overloaded to denote the set of predicate variables in state s_k^R). The first term measures the normalized difference between states up to the end of the shorter plan, while the second term, in the absence of a state to compare to, assigns maximum difference possible.

Here we illustrate the explicability distance with an example and discuss its relationship with the other distance measures. Consider the grid structure shown in Figure 2. Here we have a 5 by 5 grid. The bottom left cell is labeled as (1, 1). Some of the cells have glass floors while some others have obstacles. The robot has to find its way across the obstacles from the start cell to the goal cell. Looking at the grid structure, the human may expect the robot to take an optimal path highlighted by the green arrows. Although unbeknownst to the human, the robot has difficulty traveling across the glass floor cells because of the reflective surface and has to use special sensors while navigating across these floors. Hence, the cost of treading on these glass floors is higher than the cost of treading across normal cell. In this case, the robot's optimal plan to the goal is the one highlighted in red, which doesn't coincide with human's expectation of the robot plan.

In Figure 3, we provide the actions sets, causal link sets and state sequences generated for both the robot plan and human expected plan for our example illustrated in Figure 2. The corresponding plan distances are shown in Table I. These three distances capture different aspects of the plans. In this case, the explicability distance clearly has a high correlation with these other distance measures. Our goal in this paper

Initial State: at(1, 1)
Goal State: at(5, 4)

Actions:

$$A(\pi^R) = \{ \text{move-diagonal}(2, 2), \text{move-up}(2, 3), \text{move-up}(2, 4), \text{move-diagonal}(3, 5), \\ \text{move-diagonal}(4, 4), \text{move-right}(5, 4) \}$$

$$A(\pi^H) = \{ \text{move-diagonal}(2, 2), \text{move-diagonal}(3, 3), \text{move-diagonal}(4, 4), \text{move-right}(5, 4) \}$$

Causal Links:

$$Cl(\pi^R) = \{ \langle \text{move-diagonal}(2, 2), \text{at}(2, 2), \text{move-up}(2, 3) \rangle, \langle \text{move-up}(2, 3), \text{at}(2, 3), \\ \text{move-up}(2, 4) \rangle, \langle \text{move-up}(2, 4), \text{at}(2, 4), \text{move-diagonal}(3, 5) \rangle, \langle \text{move-diagonal}(3, 5), \\ \text{at}(3, 5), \text{move-diagonal}(4, 4) \rangle, \langle \text{move-diagonal}(4, 4), \text{at}(4, 4), \text{move-right}(5, 4) \rangle \}$$

$$Cl(\pi^H) = \{ \langle \text{move-diagonal}(2, 2), \text{at}(2, 2), \text{move-diagonal}(3, 3) \rangle, \langle \text{move-diagonal}(3, 3), \\ \text{at}(3, 3), \text{move-diagonal}(4, 4) \rangle, \langle \text{move-diagonal}(4, 4), \text{at}(4, 4), \text{move-right}(5, 4) \rangle \}$$

State sequences:

$$S(\pi^R) = \{ \{ \text{at}(2, 2) \}, \{ \text{at}(2, 3) \}, \{ \text{at}(2, 4) \}, \{ \text{at}(3, 5) \}, \{ \text{at}(4, 4) \} \}$$

$$S(\pi^H) = \{ \{ \text{at}(2, 2) \}, \{ \text{at}(3, 3) \}, \{ \text{at}(4, 4) \} \}$$

Fig. 3: The action sets, causal link sets and state sequence sets for the illustrated example in Figure 2. For the given initial and goal state, the plans illustrated in red and green in Figure 2 are used to produce respective action, causal link and state sequence sets.

TABLE I: Action distance, causal link distance and state sequence distance computed using the sets provided in Figure 3, between the two plans, π^R and π^H , that are illustrated in Figure 2.

Plan Pair	δ_A	δ_C	δ_S
(π^R, π^H)	4/7	6/7	4/5

is, then, to learn to establish a general relationship between the established measures of plan distance.

IV. PROPOSED METHODOLOGY

A. Explicability Distance

Since, without the model we do not know which plan distance is most relevant in capturing explicability, we present a general formulation in this section. A more detailed formulation can be found in the following section. Let Δ be a 3-dimensional vector, such that for a robot plan, π^R , derived from $\mathcal{M}_{\mathcal{R}}(\mathcal{R})$, and for an explicable plan π^H , derived from $\mathcal{M}_{\mathcal{H}}(\mathcal{R})$, we have $\Delta = \langle \delta_A(\pi^R, \pi^H), \delta_C(\pi^R, \pi^H), \delta_S(\pi^R, \pi^H) \rangle^T$. We now define explicability distance of a robot plan, $Exp(\pi^R)$, as a regression based function of the three plan distances, with b as the parameter vector:

$$Exp(\pi^R / \pi^H) \approx f(\Delta, b) \quad (4)$$

In order to train our regression model, we use plan traces whose actions were assigned scores by human subjects. We can then calculate the explicability score of a plan based on the average of the individual action scores.

B. Plan Generation

We now present the details of our plan generation phase, where we use the explicability distance to guide our search to generate the most explicable robot plan for a given problem.

1) *Non-Monotonicity*: We will now discuss the non-monotonic behavior exhibited by explicability function and how it affects the plan generation process. The explicability distance function is non-monotonic in nature, meaning, as the partial plan grows, the explicability distance may both increase or decrease. This is because, a new action can either contribute positively or negatively to the total explicability score of the plan. As pointed out earlier, the explicability score is computed as an average of the individual action scores in the context of the plan prefix.

Observation 1: Explicability score of a partial plan P may increase, stay equal, or even decrease when it is extended with one or more actions.

Consider the following example, in a car domain, the goal of the car is to move to the left lane. The car squeezes leftwards in three consecutive actions and after coming to the left lane, it turns on its left indicator. Here the turning on of the left tail light after having moved left is an inexplicable action. The previous three actions were explicable to the human drivers and contribute positively to the explicability

score of the plan but the last action has a negative impact and decreases the score. Therefore this score and in turn the explicability distance is not a non-decreasing function. In essence, depending on the context, the explicability of an action can either improve the score or worsen it.

Observation 2: A greedy method that expands a node with the highest explicability score of the corresponding partial plan at each step does not guarantee to find an optimal explicable plan (one of the plans with the highest explicability score) as its first solution.

The above observation is easy to see since, if e_1 is explicability score of the first plan, then a node may exist in open list (set of unexpanded nodes) whose explicability score is less than e_1 , which when expanded may result in a solution plan with explicability score higher than e_1 .

2) *Reconciliation Search:* Given the non-monotonic nature of explicability distance function, we have to generate all the candidate plans in order to find the most explicable plan. Here, we present a cost-bounded anytime greedy search algorithm called reconciliation search that generates all the valid loopless candidate solution plans up to a given cost bound, and then progressively searches for plans with better explicability scores. The value of the heuristic $h(v)$ in a particular state v encountered during search is based entirely on the explicability distance of the robot plan prefix up to that state, given by,

$$\begin{aligned} h(v) &= \text{Exp}(\pi / \pi_h) \\ \text{s.t. } \Gamma_{M_R(R)}(\mathcal{I}, \pi) &= v \\ \text{and } \Gamma_{M_H(R)}(\mathcal{I}, \pi_h) &= v \end{aligned}$$

Since we want to find explicable plans which are within a cost bound, we use the cost of the plan to prune the nodes in the search graph whenever they exceed the given maximum cost bound. We implement this search in the *Fast-Downward* planner [5]. The approach is described in detail in Algorithm 1.

At each iteration of the algorithm, the plan prefix of the robot model is compared with the explicable trace π_h (these are the plans generated by the human mental model of the robot $\mathcal{M}_H(\mathcal{R})$ up to the current state in the search process) for the given problem. Using the computed distances, we predict the explicability score for every candidate robot plan. The search algorithm then makes a locally optimal choice of states. After generating the first solution plan we do not stop the search but instead continue to find all the valid loopless candidate solution plans within the given cost bound or until the state space is completely explored. In the end, the candidate plan with highest explicability score is returned.

V. EXPERIMENTAL ANALYSIS

A. Autonomous Car Simulation Experiment

1) *Domain Model:* Autonomous cars are a topic of interest from the point of view of explicability problem. In the recent past, Google’s self-driving cars [19] have been in the news for being “too safe” on the roads. These autonomous cars governed by strict traffic rules find it hard to blend

Algorithm 1 Reconciliation Search

Input: Planning problem $\mathcal{P} = \langle M_R(R), \mathcal{I}, \mathcal{G} \rangle$, cost bound max_cost , and explicability distance function Exp

Output: Robot plan with the highest explicability score $\pi^R = \arg \max_{\pi^R} \text{Exp}(\pi^R / \pi_H)$

```

1:  $\mathcal{S} \leftarrow \emptyset$                                 ▷ Candidate plan solution set
2:  $\text{open} \leftarrow \emptyset$                           ▷ Open list
3:  $\text{closed} \leftarrow \emptyset$                       ▷ Closed list
4:  $\text{open.insert}(\mathcal{I}, 0, \text{inf})$ 
5: while  $\text{open} \neq \emptyset$  do
6:    $n \leftarrow \text{open.remove}()$                   ▷ Node with highest  $h(\cdot)$ 
7:   if  $n \models \mathcal{G}$  then
8:      $\mathcal{S}.insert(\pi \text{ s.t. } \Gamma_{M_R(R)}(\mathcal{I}, \pi) \models v)$ 
9:   end if
10:   $\text{closed.insert}(n)$ 
11:  for each  $v \in \text{successors}(n)$  do
12:    if  $v \notin \text{closed}$  then
13:      if  $g(n) + \text{cost}(n, v) \leq \text{max\_cost}$  then
14:         $\text{open.insert}(v, h(v))$ 
15:      end if
16:    else
17:      if  $h(n) < h(v)$  then
18:         $\text{closed.remove}(v)$ 
19:         $\text{open.insert}(v, h(v))$ 
20:      end if
21:    end if
22:  end for
23: end while
24: return  $\arg \max_{\pi^R \in \mathcal{S}} \text{Exp}(\pi^R / \pi_H)$ 

```

in and make judgments that would not make sense in a predominantly human environment. At four-way stops, these cars find it difficult to cross the intersection, while the human drivers keep inching forward. For a robot car, such situations, where it does not make an explicable decision can pose problems, and all the human drivers who come into contact with such cars would have to face the brunt of it.

For these reasons, we focused our studies on a simulated autonomous car environment, and investigated how the robot car’s inexplicable behavior can be avoided by generating plans with respect to their explicability scores. In our robot car model (written in PDDL), we try to capture bad driving etiquette commonly seen on roads, such as, driving below speed limit in passing lanes, overtaking from the wrong side, turning and changing lanes without showing signal, not following the move over law, and so on. The human mental model of the robot car is defined as per test subjects assumptions of how the robot car should perform actions. From the robot model $\mathcal{M}_R(\mathcal{R})$, we generated 40 plans for 16 different problems. The plans consisted of both explicable and inexplicable robot car behaviors. These plans were assessed by 20 test subjects, with each subject evaluating 8 plans. Also, each plan was evaluated by 4 different subjects, in order to get a general understanding of the assumptions of different human drivers. Therefore, the overall number of

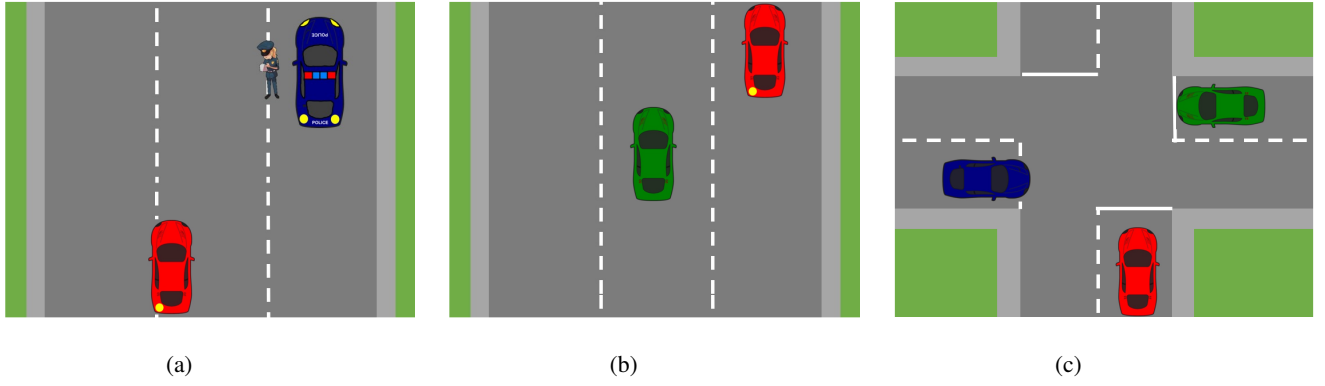


Fig. 4: Autonomous Car Domain Simulation. Here the red colored car in the images is the robot car and the rest of the cars are assumed to be human drivers. (a) The cop car is parked on the rightmost lane, and the robot car is following through the Move Over Law maneuver. (b) The robot car is wrongly trying to overtake from the rightmost lane. (c) The robot car is waiting at a four-way stop intersection even though it is the turn of the robot car to cross over.

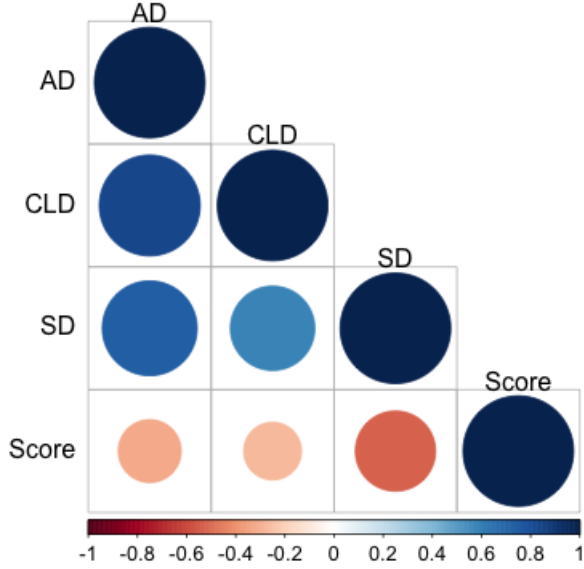


Fig. 5: Here AD, CLD, SD and Score represent action distance, causal link distance, state sequence distance, and explicability scores respectively. This is a correlation matrix for the aforementioned metrics. The red color represents the negative correlation that exists between the distance measures and the scores.

training samples was 160. The test subjects were required to have sufficient real-life driving experience. The assessment had two parts: one part involved scoring each robot car action with 1, if explicable, and 0 otherwise (the explicability score of the overall plan is calculated as the fraction of actions in the plan that were labeled as explicable); the other part involved answering a questionnaire aimed at understanding test subject’s assumptions regarding the robot car. The information from this questionnaire was used to design the human mental model $\mathcal{M}_{\mathcal{H}}(\mathcal{R})$ of the robot car.

The PDDL domain of the robot car, $\mathcal{M}_{\mathcal{R}}(\mathcal{R})$, consists of lane and car objects as shown in Figure 4. The red car

is the robot car in the experiments and all other cars seen in the experiments are assumed to have human drivers. The car objects are associated with predicates defining the location of a car on a lane segment, status of left and right turn lights, status of car being within speed limit, presence of a parked cop car, and so on. The actions possible in the domain are with respect to the robot car. These actions are Accelerate, Decelerate, LeftSqueeze, RightSqueeze, LeftLightOn, LeftLightOff, RightLightOn, RightLightOff, SlowDown and WaitAtStopSign, and so on. In order to change a lane, three consecutive actions of either LeftSqueeze or RightSqueeze are required to gradually move to the other lane. The PDDL domain of the human mental model, consists of same state predicates, but different action representations, preconditions, effects and action-costs. Note that even though representing the human mental model in PDDL may seem like a strong assumption, we validated the labels given by the human subjects with the PDDL human model constructed from the elicited preferences and found about 72.3% match, which indicates that $\mathcal{M}_{\mathcal{H}}(\mathcal{R})$ used in the evaluations is a good approximation of the human mental model of the robot

2) *Defining the Explicability Distance:* For the 22 training problems, explicable plans with $\mathcal{M}_{\mathcal{H}}(\mathcal{R})$ were generated. Since some actions are not common to both the domains and also owing to the difference in the effects and preconditions of the actions across domains, an explicit mapping was defined between the actions over the two domains. This mapping was done in the light of the plan distance operations performed between plans in the two domains.

The correlation matrix in Figure 5 establishes the negative correlation of the plan distance measures to the explicability scores. From the correlation matrix it can be seen that, causal link distance has significant negative correlation with the explicability scores. After establishing the negative correlation, we proceed towards training our regression model called explicability distance.

TABLE II: Parameters of Regression Models

Distance	b	w	Accuracy %
δ_A	0.72	-0.33	10.14
δ_C	0.73	-0.231	7.06
δ_S	0.92	-0.519	27.47
$\delta_A, \delta_C, \delta_S$	0.93	0.207, -0.061, -0.626	28.02

$$s_1^R = b_1 + w_1 \delta_A \quad (5)$$

$$s_2^R = b_2 + w_2 \delta_C \quad (6)$$

$$s_3^R = b_3 + w_3 \delta_S \quad (7)$$

$$s_4^R = b_4 + w_4 \delta_A + w_5 \delta_C + w_6 \delta_S \quad (8)$$

At first, individual distances were used to fit the data in the regression model. This resulted in a poorly learned regression model. A linear combination of the three distances also resulted in poor results. For regression model functions 5, 6, 7 and 8, the bias, weight and accuracy values were as shown in Table II. From this table, we infer that the relationships are not necessarily linear as we speculated previously. We improve our model using Random Forest regression. Since random forests allow selection of random subset of features while splitting the decision node, the accuracy of our model improves. All the three distances have statistically significant contribution in the fitted model. We evaluate the goodness of the fit of the model, using the coefficient of determination or R^2 . This value determines the measure by which the fitted model can explain the variations in the target values. This value lies between 0 to 1. Higher the R^2 value, better is the model fitted to the data. After training process the new regression model was found to have 0.8721 R^2 value. That is to say, 87% of the variations in the features can be explained by our model. Our model predicts the explicability distance between the robot plans and human mental model plans, with a high accuracy. We call this plan distance regression model as the explicability distance.

3) *Evaluation*: For evaluation of our system, we tested it on 13 different problems. We ran the algorithm with a high cost bound, in order to cover the most explicable candidate plans for all the problems. The results of this search process are as shown in Figure 6, 7 and 8. From these results, we can see that the reconciliation search is able to incrementally develop plans with better explicability scores as shown in Figure 6. In Figure 7, we see that for all the 13 problems the explicability score of the optimal plans is lesser than the final plans generated by reconciliation search. From Figure 8, we see that for the first six problems the optimal and explicable plans have same cost but our modified planner with reconciliation search produces explicable plan versions for those problems. The results also clearly show that the explicable plans can be costlier than plans that are optimal with respect to the robot's own model. This additional cost

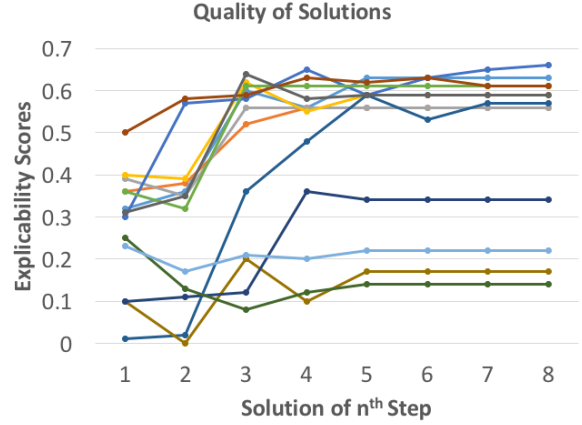


Fig. 6: The graph shows that the search process finds plans with incrementally better explicable scores. Each color line represents the 13 different problems. The markers on the lines represent a plan solution for that problem. The y-axis gives the explicability scores of the plans and the x-axis gives the solution number. Note that the curves show the non-monotonic nature of evaluation metric in the search process. The final output of the algorithm is, of course, the best plan found in the search process.

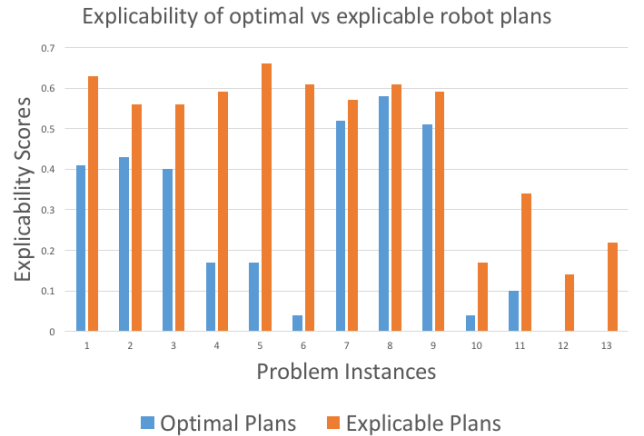


Fig. 7: For the test problem instances, the optimal plans generated using Fast-Downward planner and the plans generated using *Reconciliation Search* were compared for their explicability scores.

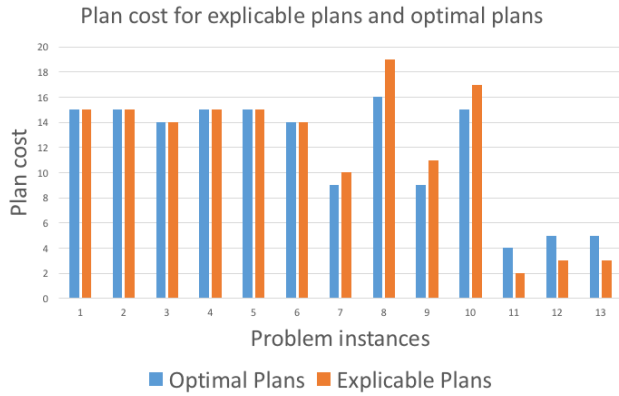


Fig. 8: For the test problem instances, the optimal plans generated using *Fast-Downward* planner and the plans generated using *Reconciliation Search* were compared for their plan costs.

can be seen as the price the robot pays to make its behavior explicable to the human.

VI. CONCLUSION

We showed how the plan distance measures play a role in determining the explicability of a robot plan. We evaluated our hypothesis in the simulated *Autonomous Car* PDDL domain. We generated training samples in robot's domain and assigned them with human scores. We also generated plans in the human's model to find the distances between plans in two domains. We looked at the relationships between scores and the distance measures of the plans. We learned the regression model that could best capture the explicability of the training samples. In summary, we have proved our hypothesis that using the human's mental model of the robot model we can assess the explicability of a robot plan as a function over the plan distance measures between the robot plan and the plan that the human would expect the robot to make. We also showed that the explicability distance measure can be used to bias the robots planning process to generate plans that are more in concordance with what humans expect. We are currently in the process of incorporating this theory into the behavior of a Fetch robot involved in delivery tasks, to demonstrate how it improves the explicability of the robot's behavior.

REFERENCES

- [1] E. de Visser and R. Parasuraman, "Adaptive aiding of human-robot teaming effects of imperfect automation on performance, trust, and workload," *Journal of Cognitive Engineering and Decision Making*, vol. 5, no. 2, pp. 209–231, 2011.
- [2] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "Pddl-the planning domain definition language," 1998.
- [3] B. Srivastava, T. A. Nguyen, A. Gerevini, S. Kambhampati, M. B. Do, and I. Serina, "Domain independent approaches for finding diverse plans," in *IJCAI*, 2007, pp. 2016–2022.

- [4] T. A. Nguyen, M. Do, A. E. Gerevini, I. Serina, B. Srivastava, and S. Kambhampati, "Generating diverse plans to handle unknown and partially known user preferences," *Artificial Intelligence*, vol. 190, no. 0, pp. 1 – 31, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0004370212000707>
- [5] M. Helmert, "The fast downward planning system," *CoRR*, vol. abs/1109.6051, 2011. [Online]. Available: <http://arxiv.org/abs/1109.6051>
- [6] H. A. Kautz and J. F. Allen, "Generalized plan recognition," in *AAAI*, vol. 86, no. 3237, 1986, p. 5.
- [7] M. Ramirez and H. Geffner, "Probabilistic plan recognition using off-the-shelf classical planners," in *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI 2010)*. Citeseer, 2010, pp. 1121–1126.
- [8] T. Chakraborti, Y. Zhang, D. E. Smith, and S. Kambhampati, "Planning with resource conflicts in human-robot cohabitation," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 1069–1077.
- [9] M. Cirillo, L. Karlsson, and A. Saffiotti, "Human-aware task planning for mobile robots," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, June 2009, pp. 1–7.
- [10] T. Chakraborti, G. Briggs, K. Talamadupula, Y. Zhang, M. Scheutz, D. Smith, and S. Kambhampati, "Planning for serendipity," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.
- [11] K. Talamadupula, G. Briggs, T. Chakraborti, M. Scheutz, and S. Kambhampati, "Coordination in human-robot teams using mental modeling and plan recognition," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 2957–2962.
- [12] S. J. Levine and B. C. Williams, "Concurrent plan recognition and execution for human-robot teams," in *ICAPS*, 2014.
- [13] Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, and S. K. Hankz Hankui Zhuo, "Plan explainability and predictability for cobots," *CoRR*, vol. abs/1511.08158, 2015. [Online]. Available: <http://arxiv.org/abs/1511.08158>
- [14] A. Dragan and S. Srinivasa, "Generating legible motion," in *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.
- [15] T. W. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, 2003.
- [16] G. Hoffman and C. Breazeal, "Cost-based anticipatory action selection for human-robot fluency," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 952–961, 2007.
- [17] Y. Zhang, S. Sreedharan, and S. Kambhampati, "Capability models and their applications in planning," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1151–1159.
- [18] R. P. Goldman and U. Kuter, "Measuring plan diversity: Pathologies in existing approaches and a new plan distance metric," 2015.
- [19] M. Richtel and C. Dougherty, "Google's driverless cars run into problem: Cars with drivers," *The New York Times*, vol. 9, p. 1, 2015.